ITRI 615

Pfleeger Chapter 5 Part 2

A trusted operating system

We say that an operating system is trusted if we have confidence that it provides

1.Memory protection,

2.File protection,

3.General object access control, and

4.User authentication

Consistently and effectively.

A trusted operating system *cont.*

The 4 major underpinnings of a trusted operating system are:

1.Policy. Statements of what the system should do and how it should do it.

2.Model. A representation of the policy the operating system will enforce. I.e. the proposed system has to meet its requirements while protecting appropriate objects and relationships.

3.Design. Involves both what the trusted operating system is and how it is to be constructed.

4.Trust. Rooted in two aspects, features (has all the necessary functionality) and assurance (confident that it will enforce the security policy correctly).

Trusted software

- We say that software is trusted software if we know that the code has been rigorously developed and analyzed, giving us reason to trust that the code does what it is expected to do and nothing more.

- Trust is based on looking for certain key characteristics:

1. Functional correctness. Program does what it is supposed to do.
2. Enforcement of integrity. Maintains the correctness of the data even with invalid commands.
3. Limited privilege. Access to secure data is minimized and not passed on.
4. Appropriate confidence level. Rated at a degree of trust appropriate for the kind of data and environment in which it will be used.

Trusted operating system design

- Good design principles are always good for security.

- Several important design principles that are quite particular to security and essential for building a solid, trusted operating system include:

1. Least privilege. Each entity should operate using the fewest privileges possible.

2. Economy of mechanism. The design of a protection system should be small, simple and straightforward.

3. Open design. The mechanism should be public, depending on the secrecy of relatively few key items.

Trusted operating system design *cont.*

4. Complete mediation. Every access attempt must be checked.

5. Permission based. The default should be denial of access, rather than the other way round.

6. Separation of privilege. Access to objects should depend on more than one condition.

7. Least common mechanism. Shared objects provide potential channels for information flow. Physical or logical separation reduce the risk from sharing.

8. Ease of use. If a mechanism is easy to use, it is unlikely to be avoided.

Security features of trusted operating systems

- The key security features of a trusted operating system include:

1. User identification and authentication.
2. Mandatory access control.
3. Discretionary access control.
4. Object reuse protection.
5. Complete mediation.
6. Trusted path.
7. Audit.
8. Audit log reduction.
9. Intrusion detection.

Assurance in trusted operating systems

- Typical operating system flaws can be attributed to four main known vulnerabilities:

1. User interaction. The largest single source.

2. Ambiguity in access policy. We want to separate users and their resources, but more often than not these users depend on shared resources.

3. Incomplete mediation. When a requested access is only authorized once per user interface operation, process execution or machine interval.

4. Generality. Operating systems tend to be too general to allow interoperability with software from other vendors.

Assurance in trusted operating systems *cont.*

- Once we understand the potential vulnerabilities in a system, we can apply assurance techniques to seek out the vulnerabilities and mitigate or eliminate their effects.

- The three available techniques are testing, verification and validation.

1. Testing. The most widely accepted assurance technique. Conclusions from testing are based on the actual product being evaluated, but unfortunately testing is almost always constrained by a project's budget and schedule. A testing strategy often used in computer security is called penetration testing, tiger team analysis, or ethical hacking.

Assurance in trusted operating systems *cont.*

2. Verification. The most rigorous method of analyzing security. Uses rules of mathematical logic to demonstrate that a system has certain security properties. Verification confirms that the operating system provides the security features it should and nothing else. The two principal difficulties of verification methods are time and complexity.

3. Validation. A more general approach to ensuring correctness. The counterpart of verification which assures that the system developers have implemented all requirements. Validation of an operating system can take place through either:

- Requirements checking.
- Design and code reviews.
- System testing.

Open source software

- What are the advantages of open source for computer security?

- What are the disadvantages of open source for computer security?

- Additional benefits of open source:

1. Cost. As the source code is available to the public, high fees will simply lead to unofficial trading.
2. Quality. The code can be analyzed by many independent reviewers.
3. Support. As the public finds flaws, they are also in the best position to suggest fixes for the flaws.
4. Extensibility. Code can easily be extended to meet new needs, and the new code can also be easily distributed.

Questions

- Any questions?